



CIBERTEC

Pruebas de Software

Equipo de Profesores



CIBERTEC

Introducción a TDD

Fundamentos de Pruebas de Software

¿Qué es TDD?

- Técnica de Aprendizaje
 - Iterativa e Incremental
 - Constructivista
- Basada en Feedback Inmediato
- Que recuerda todo lo aprendido
- Y permite asegurarnos de no haber “desaprendido”
- Incluye análisis, diseño, programación y testing

¿Qué es TDD?

- Técnica de Desarrollo basada en “testear” primero
 - Es más que una técnica de testing
 - Es más que una técnica de programación
 - Es más que una técnica de diseño
 - Es todo junto
- Surge como una de las técnicas de XP

¿Cómo se hace TDD?

1. Escribir un test
 - Debe ser el más sencillo que se nos ocurra
 - Debe fallar al correrlo
2. Correr todos los tests
 - Si hay errores, implementar lo mínimo necesario para que pasen y GOTO 2
3. Reflexiono - ¿Se puede mejorar el código?
 - Sí -> Refactorizar. GOTO 2
 - No -> GOTO 1

¿Qué no es TDD?

- No es Testing (únicamente)
- No es Unit Testing
- No es “no reflexionar” – “no pensar”
- No es “no refactorizar”

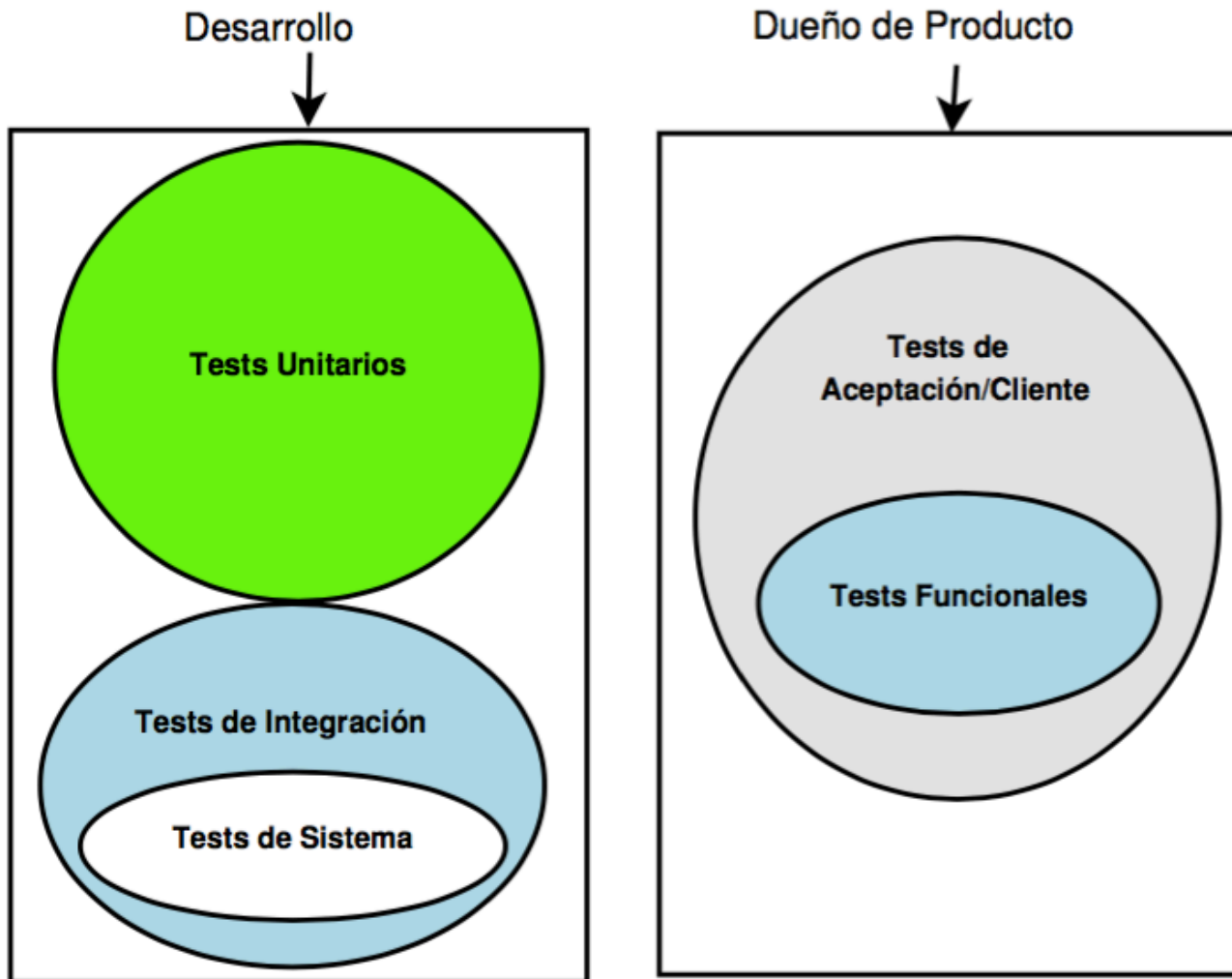
¿Cuándo no hago TDD?

- Cuando no tengo feedback inmediato
 - Escribiendo/Modificando código antes de escribir un test
- Escribiendo muchos tests antes de escribir el código

¿Cuándo no hago TDD?

- Cuando no desarrollo de manera iterativa-incremental
 - Escribiendo la “solución completa” de entrada
 - Haciendo up-front design
 - Escribiendo los tests luego de tener el código – Haciendo Testing

Tipos de Pruebas



Framework de Testing: xUnit

- Implementado por Kent Beck
- Framework de caja blanca pero simple
- Los casos de test se implementan en métodos, ej: testSimpleAdd
- Los casos de test se agrupan en clases (suites “naturales”), ej: MoneyTest
- El Test Runner usa reflexión para saber que tests correr
- Primera implementación: SUnit
- Nombre es “misleading”